

TITLE

OBJECT MANAGEMENT SYSTEM AND METHOD

BACKGROUND OF THE INVENTION

Field of the Invention

5 The present invention relates to an object management system and method, and particularly to a system and method of managing object exchange between entities via a platform.

Description of related Art

10 With the wide use of computer systems, specific functions and systems are developed for different tasks. For example, a computer host of an enterprise can comprise a workflow management system, a manufacturing planning system, an enterprise resource planning (ERP) system and others to perform related operations inside the enterprise.

15 Since these systems may cooperatively work and use common data, the systems often need to exchange objects or messages. Fig. 1 illustrates object exchange in a conventional system. As shown in Fig. 1, if all systems (system A 11, system B 12, system C 13, system D 14 and system E 15) need to exchange objects, the message flow is complex. If more systems are installed in the computer host, the object exchange between systems will become more complicated.

20 Further, when systems of different computer hosts need to exchange objects through a network, the bandwidth of the network is significantly reduced, resulting in instability of cooperation. In addition, when systems are removed from or added to the work environment, the object exchange interface and mechanism of other systems communicating therewith require

redesign, thereby increasing the difficulty of system design, expansion and maintenance.

SUMMARY OF THE INVENTION

5 The present invention provide a system and method of managing object exchange between entities via a platform.

The present invention provides an object management system and method. The system includes an object access interface, a resource management module, an access management module and an object management module.

10 The object access interface receives an access request, from at least one entity, including at least identification of the entity, a source port, object information and access content. The resource management module assigns a command handler to the access request. The access management module
15 determines authority for the access request according to the identification and the source port. The object management module performs a procedure corresponding to the access content on the object designated by the object information. After the procedure is finished, the command handler is released.

20 In addition, if the resource management module has no available command handler might be assigned to the access request, the access request is rejected or waits until one command handler is available.

25 The object management method according to the present invention first receives an access request from an entity, wherein the access request includes at least identification of the entity, a source port, object information and access content. Then, a command handler is assigned to the access request. Thereafter, authority for the access request is

determined according to the identification and the source port.
After, a procedure corresponding to the access content is
performed on the object designated by the object information.
After the procedure is finished, the command handler is
5 released.

BRIEF DESCRIPTION OF THE DRAWINGS

The aforementioned objects, features and advantages of the
invention will become apparent by referring to the following
detailed description of the preferred embodiment with reference
10 to the accompanying drawings, wherein:

Fig. 1 is a schematic diagram illustrating object exchange
in a conventional system;

Fig. 2 is a schematic diagram illustrating object exchange
according to the present invention;

15 Fig. 3 is a schematic diagram illustrating the system
architecture of the object management system according to the
present invention;

Fig. 4 is a schematic diagram illustrating the relationship
between access ports according to the invention;

20 Fig. 5 is a schematic diagram illustrating the components
of the object management module according to the present
invention;

Fig. 6 is a flowchart showing the process of object writing
according to the present invention;

25 Fig. 7 is a flowchart showing the process of object reading
according to the present invention;

Fig. 8 is a flowchart showing the process of object removal
according to the present invention; and

Fig. 9 is a flowchart showing the process of object retrieval according to the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Fig. 2 is a schematic diagram illustrating object exchange according to the present invention. As shown in Fig. 2, entities 201, 202, 203 and 204 may use the object management system 200 of the present invention as a platform to exchange and transmit objects. It should be noted that the entities might be users, application systems, software components, or others. Compared to Fig. 1, the present invention may significantly reduce message flow between entities, and thus the complexity of the work environment is reduced.

Fig. 3 illustrates the system architecture of the object management system according to the present invention.

The object management system 200 according to the present invention includes a local interface 210, a remote interface 220, an object access interface 230, a resource management module 240, an access management module 250, authority storage 260, an object management module 270 and object storage 280.

The local interface 210 provides access service for at least one local entity 310, such those on the computer host of the object management system 200. The local entity 310 may directly use resources, such as memory and hard disk of the computer host. The remote interface 220 provides access for at least one remote entity 320, in which the remote entity 320 communicates with the object management system 200 via a network. At the time, the remote interface 220 is responsible for related operations of network transmission, such as packet combination and disassembly.

After the local interface 210 and the remote interface 220 receive the access request from the local entity 310 and the remote entity 320, and perform related operations, the received access request is forwarded to the object access interface 230.

5 The object access interface 230 may be an interface to handle access requests from the local entity 310 and the remote entity 320.

The access request may include identification of the entity, a source port (source access port), object information, access content, and a destination port, selectively. It should be noted that each entity might have a plurality of access ports, each access port having respective authority for transmitting objects to a specific access port. For example, in Fig. 4, the entity A 410 owns the access port A 411 and the access port B 412, the entity B 420 owns the access port C 421 and the access port D 422, in which the access port A 411 has the authority to transmit objects to the access port C 421 and the access port D 422, the access port B 412 has the authority to transmit objects to the access port C 421, the access port C 421 has the authority to transmit objects to the access port A 411 and the access port B 412, and the access port D 422 has the authority to transmit objects to the access port A 411.

The resource management module 240 manages the resources of the computer host, the module 240 includes a predetermined number of command handlers, and assigns one thereof to the access request if the access request is received by the object access interface 230. After the procedure corresponding to the access request is finished, the command handler is released. It should be noted that if all of the command handlers are assigned to different access requests, a new access request received by the

object access interface 230 is rejected or waits. Once one assigned command handler is released, the resource management module 240 may assign the released command handler to the new access request.

5 The access management module 250 manages authority for the object management system 200, each access port, and object transmission between the access ports for each entity. The authority information is stored in the authority storage 260.

10 The object management module 270 may perform procedures corresponding to the access content on the object designated by the object information according to authority for the access request. Fig. 5 illustrates the components of the object management module 270 according to the present invention. As shown in Fig. 5, the object management module 270 at least
15 includes an object pool 271, an object ID (identification) generation unit 272, an object lock/release unit 273, a sync unit 274 and an object storage access interface 275.

20 All objects in the object management system 200 are stored in the object pool 271, in which the object pool 271 may be the memory of the computer host. When a new object enters the object management system 200, the object ID generation unit 272 generates a unique ID for the object for access identification and determination. After the object is retrieved by one entity, the object lock/release unit 273 locks the object, such that the
25 other entities cannot access the locked object (in other words, the object is not available for other entities). After the entity finishes handling the object or a preset time passes, the object lock/release unit 273 releases the object. If the work environment includes several object management systems, the
30 sync unit 274 may synchronously update the variation of the

objects in the object pool 271 to the other object management systems. The object storage access interface 275 is an interface for the object management module 270 to access the object storage 280.

5 Fig. 6 shows the process of object writing according to the present invention. First, in step S601, the object access interface 230 receives an access request from the entity (local entity 310 or remote entity 320) via the local interface 210 or the remote interface 220. The access request includes
10 identification of the entity, a source port, object information, access content including a write command, and a destination port. Then, in step S602, the resource management module 240 checks whether any command handler is available. If all of the command handlers have been assigned (No in step S602), the flow
15 stays in step S602 until a command handler is released. Otherwise (Yes in step S602), in step S603, the resource management module 240 assigns a command handler to the access request.

 Thereafter, in step S604, the access management module 250
20 retrieves authority (or determines authority) for the access request from the authority storage 260 according to the identification of the entity, the source port and the destination port. If the access request is not permitted (No in step S605), in step S606, the access request of the entity
25 is rejected. Otherwise (Yes in step S605), in step S607, the object ID generation unit 272 generates object ID for the object corresponding to the object information, and the object management module 270 writes the object into the destination port in the object pool 271. At the same time, in step S608,

Client's ref.: VIT03-0190
Our ref.: 0608-10149-USf/Yianhou/Kevin

the sync unit 274 synchronously duplicates the object to the other object management systems.

It should be noted that if the access content includes a permanent write command, the object management module 200 further writes the object to the object storage 280. After the write operation of the object finishes, in step S609, the assigned command handler is released.

Fig. 7 shows the process of object reading according to the present invention. First, in step S701, the object access interface 230 receives an access request, from the entity (local entity 310 or remote entity 320) via the local interface 210 or the remote interface 220, including identification of the entity, a source port, object information, and access content including a read command. Then, in step S702, the resource management module 240 checks whether any command handler is available. If all of the command handlers have been assigned (No in step S702), the flow stays in step S702 until a command handler is released. Otherwise (Yes in step S702), in step S703, the resource management module 240 assigns a command handler to the access request.

Thereafter, in step S704, the access management module 250 retrieves authority for the access request from the authority storage 260 according to the identification of the entity and the source port. If the access request is not permitted (No in step S705), in step S706, the access request of the entity is rejected. Otherwise (Yes in step S705), in step S707, the object corresponding to the object information is provided to the entity for reading. After the read operation of the object finishes, in step S708, the assigned command handler is released.

Fig. 8 shows the process of object removal according to the present invention. First, in step S801, the object access interface 230 receives an access request from the entity (local entity 310 or remote entity 320) via the local interface 210 or the remote interface 220, including identification of the entity, a source port, object information, and access content including a remove command. Then, in step S802, the resource management module 240 checks whether any command handler is available. If all of the command handlers have been assigned (No in step S802), the flow stays in step S802 until a command handler is released. Otherwise (Yes in step S802), in step S803, the resource management module 240 assigns a command handler to the access request.

Thereafter, in step S804, the access management module 250 retrieves authority for the access request from the authority storage 260 according to the identification of the entity and the source port. If the access request is not permitted (No in step S805), in step S806, the access request of the entity is rejected. Otherwise (Yes in step S805), in step S807, the object corresponding to the object information is removed. At the same time, in step S808, the sync unit 274 synchronously notifies the other object management systems to remove the object. After the remove operation of the object finishes, in step S809, the assigned command handler is released.

Fig. 9 shows the process of object retrieval according to the present invention. First, in step S901, the object access interface 230 receives an access request from the entity (local entity 310 or remote entity 320) via the local interface 210 or the remote interface 220, including identification of the entity, a source port, object information, and access content

including a retrieve command. Then, in step S902, the resource management module 240 checks whether any command handler is available. If all of the command handlers have been assigned (No in step S902), the flow stays in step S902 until a command handler is released. Otherwise (Yes in step S902), in step S903, the resource management module 240 assigns a command handler to the access request.

Thereafter, in step S904, the access management module 250 retrieves authority for the access request from the authority storage 260 according to the identification of the entity and the source port. If the access request is not permitted (No in step S905), in step S906, the access request of the entity is rejected. Otherwise (Yes in step S905), in step S907, the object corresponding to the object information is provided to the entity. At the same time, in step S908, the object lock/release unit 273 locks the object, disabling access from other entities. It should be noted that if the entity finishes handling the object, the object lock/release unit 273 releases the object. In addition, if the access request has a preset time limit, the object lock/release unit 273 releases the object after the preset time. After the retrieval operation of the object finishes, in step S909, the assigned command handler is released.

As a result, the object management system and method according to the present invention provides object exchange and transmission between systems or entities via a platform, thereby reducing message flow and complexity of the work environment. In addition, when systems are removed from or added to the work environment, the object exchange interface and mechanism of existing systems does not require removal or redesign, thereby

Client's ref.: VIT03-0190
Our ref.: 0608-10149-USf/Yianhou/Kevin

reducing the difficulty of system design, expansion and maintenance.

Although the present invention has been described in its preferred embodiments, it is not intended to limit the invention to the precise embodiments disclosed herein. Those who are skilled in this technology can still make various alterations and modifications without departing from the scope and spirit of this invention. Therefore, the scope of the present invention shall be defined and protected by the following claims and their equivalents.